Detailed Research Statement—Damek Davis

1. Introduction. I design and analyze optimization algorithms for machine learning, statistics, and signal-processing problems. The optimization problems I find most exciting (e.g., those in deep learning) fall outside the scope of classical theory since they lack conventionally helpful notions of convexity or smoothness. For such problems, the most promising practical algorithms today are simple nonconvex optimization heuristics (e.g., SGD), and except for a few exceptional cases, there is no guarantee they will find global optima. Despite NP-hardness, these simple heuristics often succeed, and over the last decade, I have studied why and when they do. My work on this topic has received several awards, including the Alfred P. Sloan Fellowship in Mathematics (2020); the INFORMS Optimization Society Young Researchers Award (2019); the NSF CAREER Award (2020); and the SIAM Activity Group on Optimization Best Paper Prize (2023).

1.1. Backdrop. In the 2010s, the tech industry was just starting to routinely use simple iterative methods (e.g., backpropagation) with ad-hoc initializations to train nonsmooth neural networks formed from compositions of many Rectified Linear Units (ReLUs) $\sigma(x) = \max\{x, 0\}$. Now, we plug neural nets into every piece of the applied modeling pipeline. In response, the optimization community has had to rethink our nonconvex toolkit and depart from the prevailing approach of the past decades: formulating convex relaxations based on semi-definite programming (SDP), which are extremely powerful but not efficiently solvable in high-dimensions by current methods. Consequently, we have had to shift our focus back to directly optimizing nonconvex problems through the dynamics of simple iterative methods.

1.2. The technical challenge. The aughts brought a revolution in our understanding of first-order methods for convex optimization, culminating in fast algorithms for solving compressive sensing problems in computational imaging. However, they left many basic algorithmic questions for problems without smoothness or convexity relatively untouched. For example, do training algorithms, such as SGD, converge? If so, what do they converge to–saddle points or local minima? And how quickly do they converge? Are the existing algorithms the best possible, or can we provably accelerate them? These questions are classical in smooth and convex optimization and rely on, e.g., Taylor's theorem, the stable manifold theorem, and the relatively simple global geometry of convex functions. For problems that are nonsmooth and nonconvex, these tools break down.

1.3. Contributions. Practical deep learning training is challenging to analyze using classical tools since the best-performing models appear almost pathological—a massive web of compositions of nonlinear nonsmooth functions with trainable parameters inserted wherever the practitioner chooses. But it seems to work, so it is crucial to take it seriously so that we might improve its efficiency and reliability. And upon closer inspection, these problems are not too pathological—we build them from just a few simple components, like polynomials, exponentials, logs, max's, min's, and absolute values. They are examples of so-called "tame" functions/optimization problems, a virtually exhaustive class in applications, which includes all semialgebraic and subanalytic functions [1]. Tameness, in fact, implies beneficial "partial smoothness" properties. For example, every tame function's graph is the finite union of "manifolds" that fit together in a "regular" pattern.

This research statement summarizes a thread of my work that uncovers/exploits beneficial partial smoothness properties in order to analyze, design, and accelerate optimization algorithms for nonconvex and/or nonsmooth problems. It also summarizes a thread of my recent work on state-of-the-art LLM optimizers [2], like Muon [3], which I'm continuing to actively work on. While deep learning formulations motivate some of my work, my wider aim is to discover principles that

generalize to broad optimization problems in machine learning, statistics, and signal processing. I will describe some of my works on this theme, including:

1. An exponential acceleration of first-order methods for "generic" nonsmooth optimization problems [4] (Section 3);
2. Convergence of SGD on virtually any neural network [5] (Section 4);
3. The first efficiency guarantees for SGD for nonsmooth nonconvex problems [6, 7] (Section 5);
4. Provable recovery guarantees in nonconvex statistical-estimation problems [8–12] (Section 6);
5. A theory of "avoidable saddle-points" in nonsmooth optimization [13–15] (Sections 7 and 8);
6. The first "asymptotic normality" result for SGD in nonsmooth optimization [16] (Section 9);
7. An exponential acceleration of root-finding methods for "generic" nonsmooth mappings [17] (Section 10).
8. Identification of when and why spectral gradient methods (e.g., Muon [3]) help in deep learning [2] (Section 11).

These works are stepping stones toward establishing efficiency, generalization, and provable recovery guarantees for training/optimization methods in machine learning and signal processing. More broadly, my vision for optimization theory in the context of modern machine learning and "AI" is that it should ideally provide guidance on the model (e.g., the architecture of a neural network, interpretability of the parameters), the algorithms (e.g., "good" hyperparameter selection, "implicit bias" of generated solutions), convergence speed of methods, initialization, nonvacuous generalization guarantees based on sample size, dataset selection (e.g., removing outliers and mitigating adversarial training examples), and how to formulate tractable convex relaxations, among other topics. Unfortunately, theoretically justified recommendations in modern machine learning today often apply only in situations that are uncommon in practice. For example, the strongest guarantees available for globally optimizing the loss function in machine learning apply in extremely "wide" neural networks—the neural-tangent-kernel (NTK) regime—which are much wider than those used in practice;[1] they apply when the neural network perfectly interpolates the data, which does not appear to be true in the modern era of large-language models; they suggest using hyperparameters that perform poorly in practice (e.g., small "stepsizes" and extremely large "batch sizes"); they often ignore *generalization error*–the gold standard metric in ML; they assume "smooth activation" functions, which appear in practice, but rule out commonly used ReLUs. The difficulty of providing good recommendations that work at scale is compounded by the secrecy of top AI companies, whose strong internal algorithms are closely guarded. Besides machine learning, these issues (and others) affect any field downstream of it, e.g., in physics-based imaging where denoising techniques based on neural network "generative priors" have recently emerged as a fruitful technique; see Section 10.

In my future work, I aim to narrow this gap between optimization theory and practice, not only due to the joy of mathematical discovery but also because I foresee that scientific and societal decisions will increasingly be made based on algorithmic output. It is thus crucial to understand how these algorithms work to ensure that they function as intended and deliver accurate results

[1] In this regime, the optimization problem essentially reduces to least squares.

within a given timeframe. Efficiency, in particular, will be a serious issue for smaller organizations due to the hoarding of GPUs by large companies and regulatory regimes that raise compliance costs for high-risk and general-purpose models[2,3] and the need for constantly retrained, real-time, up-to-date models used in production. Beyond that, without a theoretical framework that explains the generalization, robustness, and "safety" of such methods, I am concerned that (i) overly restrictive regulatory measures could take hold, concentrating access to "AI techniques" in a few large corporations that do not necessarily have our best interests in mind; and (ii) that these technologies will not be "safely deployed," for example, in the transportation, legal, and defense industries, among others.

In the rest of this statement, I describe the selected research contributions mentioned above in detail. Besides these contributions to optimization theory, I have also enjoyed working with and plan to continue working with experts in applied domains, such as coherent diffraction imaging [19] (past collaboration with John Miao and Stan Osher at UCLA), air traffic management (past collaboration with NASA) [20–22], seismic imaging [23], and computer vision [24, 25] (past collaboration with Stefano Soatto at UCLA).

Three Best Papers

2. Brief Description of Three Best Papers. Of the eight papers/themes above, my three best works are [4], [6], and [14] appearing above in Items 1, 3, and 5, respectively. Each paper addresses a different stage of the dynamics of "gradient methods" in nonsmooth nonconvex problems, as illustrated in Figure 1. Such methods aim to reach the lowest point in the figure. Along the way, they may become attracted to and trapped at any of the four "first-order critical points," depicted by the red dots. From top to bottom, the first and second dots depict "saddle points," while the third and fourth depict a "local minimum" and a "global minimum," respectively.

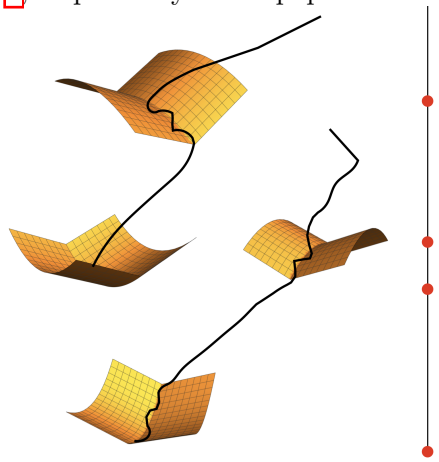


Figure 1: Landscape of a "typical" nonsmooth optimization problem together with "typical" algorithm trajectories.

A first question is how many "steps" of such methods are needed before they nearly reach the height of at least one of the four red dots. Paper [6] developed the now standard approach for estimating the number of such steps for a broad class of nonsmooth nonconvex problems, known as the "weakly convex" class. This class includes a variety of problems in statistical estimation, reinforcement, and (adversarial) machine learning problems. In recognition of the impact of this work, Paper [6] has received both the INFORMS Optimization Society Young Researchers Award in 2019 and the SIAM Activity Group on Optimization Best Paper Prize (2023). See Section 5 for further details.

Given that "gradient methods" approach the "first-order critical points" depicted in Figure 1, a second question is whether such methods reach the "good" critical points – the local or global minimum – or whether they are trapped at the saddle points. For "generic" *smooth* optimization

[2]The EU has adopted the AI Act; see the European Parliament summary.

[3]In [18, Section 2.2], Stanford statistician David Donoho presents a compelling summary of how fears of "AI killing us all"—which are influencing the policy discussion—emerged; throughout the rest of the manuscript, Donoho presents a compelling counternarrative.

problems, randomly initialized methods can never be trapped at saddle points [26, 27], a consequence of Sard's theorem and the stable manifold theorem. Nonsmooth problems are much more difficult to analyze since these tools break down. Nevertheless, Paper [14] shows that a minimal modification of standard gradient methods – simply adding a random perturbation at each step – allows one to avoid such points. See Section 7 for further details.

Once all saddle points are bypassed, gradient methods enter a region around a local or global minimizer. For "generic" smooth optimization problems, such regions have favorable structure that enable standard gradient methods to converge exponentially fast to a solution. Unfortunately, standard gradient methods can be exponentially slower for nonsmooth problems, even for "well-behaved" convex problems. Paper [4] developed the first algorithm to break this (local) complexity barrier, exponentially improving on all prior methods. In short, the algorithm's (local) "first-order oracle complexity of gradient methods" on "generic semialgebraic problems" improves the previous best complexity of $O(k^{-1/2})$ to $O(\exp(-k^{1/3}))$. See Section 3 for further details.

Details of Selected Work

3. An exponential acceleration for nonsmooth optimization. In smooth optimization, gradient methods converge linearly (i.e., exponentially fast) on functions that grow quadratically away from minimizers. Quadratic growth in turn is "generic:" if f is sufficiently smooth, almost all linear perturbations of f have quadratic growth near each local minimizer.[4] Quadratic growth is also a "generic" property of nonsmooth tame optimization problems [28]. However, since the pioneering work of Nemirovski and Yudin in the 1980s, there was thought to be an *exponential gap* between the performance of gradient methods for smooth and nonsmooth problems under this regularity condition (Figure 2) [29]. The gap already appears on the simple nonsmooth strongly convex function

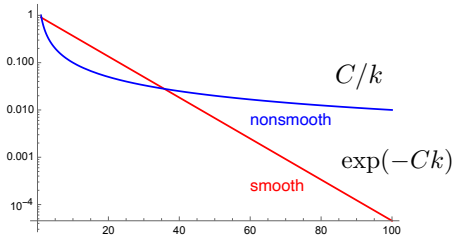


Figure 2: The speed of gradient methods under quadratic growth.

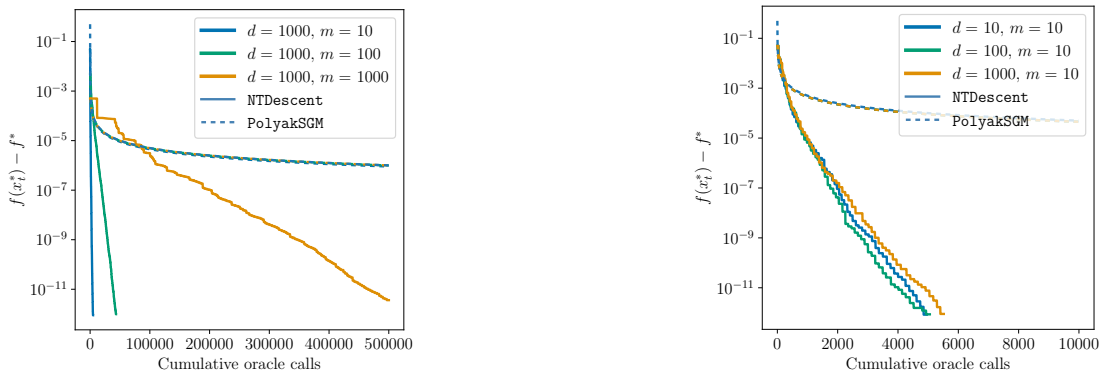
$$f(x) = \max_{1\le i\le m} x_i + \frac{1}{2}\|x\|^2 \qquad \text{for some } m \le d \text{ and all } x \in \mathbb{R}^d. \tag{1}$$

Indeed, let x_k denote the iterates of the *subgradient method*, which repeats

$$x_{k+1} = x_k - \alpha_k v_k \qquad \text{where } v_k \in \partial f(x_k), \tag{$\mathcal{SM}$}$$

where $\{\alpha_k\}$ is a control sequence and $\partial f(x)$ denotes the *subdifferential* at a point $x \in \mathbb{R}^d$ in the sense of convex analysis. Then, if one initializes x_0 at the origin and employs an "adversarial first-order oracle," there is a lower bound: $f(x_k) - \inf f \ge (2m)^{-1}$ for all $k \le m$; see [30, 31]. Beyond ($\mathcal{SM}$), the lower bound holds for any algorithm whose kth iterate lies within the linear span of the past $k-1$ subgradients. Thus, one must make more than m first-order *oracle calls* to f, i.e., function and subgradient evaluations, before possibly seeing a speedup. However, for $k \gg m$ oracle calls, existing first-order methods continue converging slowly even when given knowledge of the minimal function value $\inf f$, as in the popular Polyak stepsize (`PolyakSGM`) [32]; see dashed lines in Figure 3.

[4] A consequence of Sard's theorem.



(a) **(b)**

Figure 3: *Comparison of* `NTDescent` *with* `PolyakSGM` *on* (1). *Left: we fix d and vary m; Right: we fix m and vary d. For both algorithms, $f(x_k^*)$ denotes the best function value after k oracle evaluations.*

Recently in [4], I developed a first-order method called *Normal Tangent Descent* (`NTDescent`) that exponentially(!) surpasses the "speed limit" of gradient methods derived by Nemirovski and Yudin [29].[5] The method (locally) converges at the rate $f(x_k) - f^* = O(\exp(-C_f k^{1/3}))$ where C_f depends only on f and not the dimension of the decision variable. The guarantee applies to "almost every problem" in practice: for any tame (e.g., semialgebraic) locally Lipschitz (nonconvex) function f, almost all tilts $f(x) + \langle v, x \rangle$ for $v \in \mathbb{R}^d$ meet our assumptions. `NTDescent` is also parameter-free, so the practitioner need not set any parameters to achieve the speedup. Figure 3 illustrates the performance of `NTDescent` on f from (1). In both plots, `NTDescent` improves on `PolyakSGM`, measured in terms of oracle calls. The number of oracle calls is a fair basis for comparison since both `PolyakSGM` and `NTDescent` perform a similar amount of computation per oracle call. Figure 3b also verifies that the performance of `NTDescent` is *dimension independent.* I found this to be super surprising!

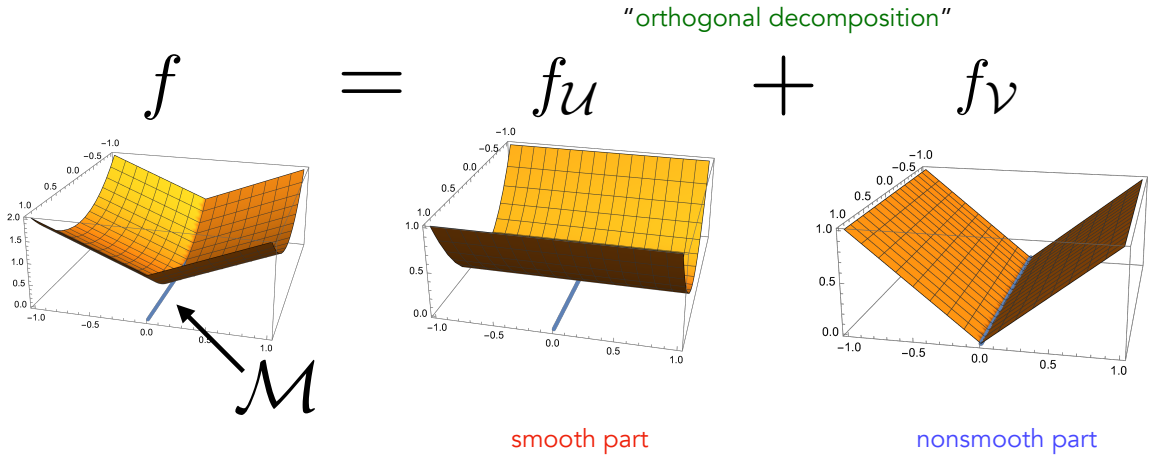


Figure 4: *An illustration of "partial smoothness" and the induced orthogonal decomposition.*

`NTDescent` is a somewhat sophisticated first-order method, so I will not state it here. But the inspiration for the method can already be gleaned from a certain "partial smoothness" property

[5] PyTorch code available at `https://github.com/COR-OPT/ntd.py`.

that induces an "orthogonal decomposition" of the loss function into smooth and nonsmooth pieces; see Figure 4. This structure was identified in my past work [14] with precursors in the work of many others [33–37]. A consequence of [14] roughly states that "generic" tame optimization problems have finitely many critical points; every local minimizer is contained within a manifold $\mathcal{M}$;[6] along the manifold, the function is smooth and has quadratic growth; in directions normal to the manifold, the function grows sharply (i.e., linearly); the manifold induces a decomposition of the function into "orthogonal pieces" consisting of the smooth extension $f_{\mathcal{U}}$ of the function from the manifold and the residual function $f_{\mathcal{V}}$. A cartoon representation of such a function is $f(u,v) = u^2 + |v|$. Once one knows the decomposition exists, one can attempt to identify the pieces and run appropriate gradient methods in the $\mathcal{U}$-space and the $\mathcal{V}$-space. For example, consider $f(u,v)$. Gradient descent on u^2 with a constant stepsize converges linearly. On the other hand, gradient descent on $|v|$ with a geometrically decaying stepsize converges linearly. Of course, one has no information about $\mathcal{M}$ nor the decomposition, so this is only a conceptual method. Nevertheless, one can think of `NTDescent` as an approximate implementation of this strategy.

4. A baseline training guarantee for all neural networks. `NTDescent` exponentially improves on existing first-order methods. However, it is still worthwhile to understand what, if any, guarantees one can provide the most common training algorithms, for example, the so-called stochastic subgradient method (SGD). SGD is a variant of ($\mathcal{SM}$) where only stochastic estimates of v_k are available, either due to inherent uncertainty in evaluating f or induced by the practical difficulty of fully evaluating derivatives of sums of loss functions over large datasets.

In [5], I proved that SGD converges to a first-order stationary point on any locally Lipschitz tame loss function f. To show this result, we uncovered a "partial smoothness" property of tame functions: like the gradient vector field for a smooth function, subdifferentials[7] of tame losses are "conservative" [39], meaning their (Aumann) integral along any arc $x\colon \mathbb{R}_+ \to \mathbb{R}^d$ coincides with the difference of function values at the arc's endpoints:

$$\{f(x(t)) - f(x(0))\} = \int_0^t \langle \partial f(x(s)), \dot{x}(s)\rangle ds$$

We then use this to show that the continuous-time analog of the algorithm $-\dot{x}(t) \in \partial f(x(t))$ is well-behaved, opening the door to classical tools from stochastic approximation theory (e.g., the ODE method). In contrast to prior work, which assumed particular architectures/data sets, this result endows all modern neural network models with asymptotic training guarantees. However, it shows limit points are critical asymptotically, and does not provide finite time efficiency guarantees.

5. Efficiency of Stochastic Methods for Convex Composite Problems. Is it possible to establish nonasymptotic guarantees? In general, efficiency estimates for SGD and other standard training methods appear out of reach under such weak "nonpathological" assumptions. Instead, in the works [6, 7], I gave the first nonasymptotic efficiency estimates for a wide class of *stochastic methods*, which apply to the ubiquitous class of *convex composite losses*. Such losses are the composition of an (outer) nonsmooth convex function h and an (inner) smooth nonlinear map c, and thus encompass, for example, a broad family of signal processing problems with

[6] In the context of (1), the manifold is the subspace in which the first m variables take on the same value: $\mathcal{M} = \{x \in \mathbb{R}^d \colon x_1 = x_2 = \dots x_m\}$.

[7] A set-valued generalization of the gradient comprised of limiting convex combinations of gradients at nearby points [38]. In classical circumstances, the subdifferential reduces to more familiar objects. For example, when f is C^1-smooth at x, the subdifferential $\partial f(x)$ consists only of the gradient $\nabla f(x)$, while for convex functions, it reduces to the subdifferential in the sense of convex analysis.

smooth "measurement" mappings c, fit by nonsmooth (e.g., ℓ_1) penalization $h \circ c$ of residuals. More generally, when problem data z follows a fixed unknown distribution $\mathcal{P}$, the proposed stochastic algorithms attempt to minimize the expected residual $f(x) = \mathbb{E}_{z\sim\mathcal{P}}[h(c(x,z),z)]$ as follows: at each iteration t, draw a sample $z_k \sim \mathcal{P}$, replace the loss $h(c(y,z),z)$ with a *local convex model* $f_{x_t}(y,z_k)$, and choose x_{k+1} to minimize $f_{x_k}(y,z_k) + \frac{1}{2\alpha_t}\|y-x_k\|^2$. For example, classical stochastic subgradient methods have this form with *linear* $f_{x_k}(y,z_k)$, explaining their notorious sensitivity to the (user selected) stepsize α_k, since linear functions poorly approximate $h(c(x,z),z)$.

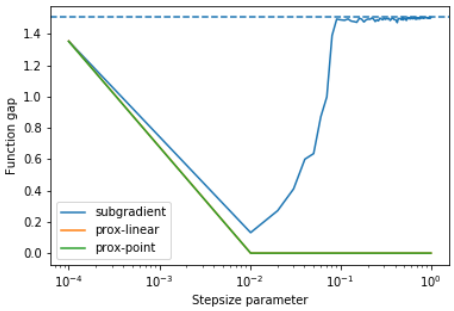
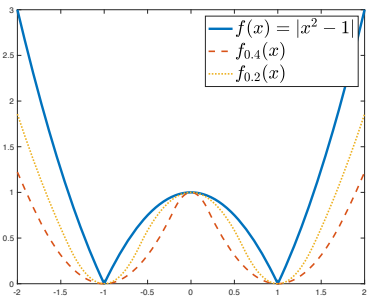


Figure 5: Step-size insensitivity of nonlinear model-based methods (prox-linear, prox-point). Shown: Function gap after 100 data passes on a phase retrieval problem.

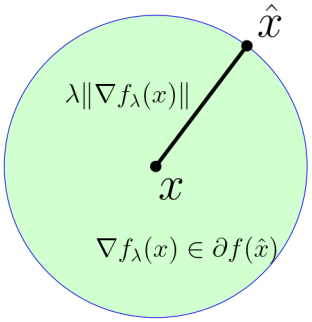
In these works [6, 7], I prove the first nonasymptotic efficiency estimates for both the classical stochastic subgradient method and a broad class of nonlinear *model-based* algorithms, for example, those generated by the *prox-linear* model $f_{x_k}(y,z_k) = h(c(x) + \nabla c(x)(y-x))$. As Figure 5 illustrates, nonlinear models empirically depend less on "stepsize tuning" than subgradient methods, but importantly, I prove they share the same efficiency estimates. More broadly, the work overcame what was thought to be a fundamental barrier to understanding streaming algorithms in nonsmooth nonconvex optimization: the conventional measures of algorithm progress, namely the objective gap and the norm of the gradient, can be completely meaningless. Indeed, on the one hand, one cannot expect the objective gap $f(x_k) - \inf f$ to tend to zero, even in a smooth setting. On the other hand, simple examples, e.g., $f(x) = |x|$, show that any "gradient" norm can be strictly bounded below by a fixed constant for all k. My work offers a surprising resolution: the gradient of an "implicit smoothing" (the *Moreau envelope* [40]) tends to zero along the iterate sequence and bounds the distance of the current iterate to a nearby point with "small" subgradient. Figure 6a plots the Moreau envelope of a simple loss function, which is defined for $\gamma > 0$ as

$$f_\gamma(x) = \min_y \left\{ f(y) + \frac{1}{2\gamma}\|y-x\|^2 \right\}. \qquad (\mathcal{MOREAU})$$

Figure 6b illustrates the relationship between gradients of f_γ and f, namely the gradients of f_γ are subgradients of f at the point argmin $\hat{x}$ of ($\mathcal{MOREAU}$) and the distance to $\hat{x}$ is $\lambda\|\nabla f_\lambda(x)\|$.



(a) *Moreau envelope of* $f(x) = |x^2 - 1|$



(b) *Approximate stationarity*

Figure 6: *An illustration of the Moreau envelope*

This work has found broad applicability in stochastic optimization, distributionally robust optimization, statistical learning, reinforcement learning, and so-called minimax optimization problems

arising in games; e.g., [41–59]. In recognition of the impact, it has received both the INFORMS Optimization Society Young Researchers Award in 2019 and the SIAM Activity Group on Optimization Best Paper Prize (2023).

6. Rapid Local Convergence in Statistical Recovery Problems. Generally, one can at most hope to find first-order stationary points of nonconvex losses, not global minima. However, medium-scale numerical tests demonstrate simple local search heuristics' remarkable ability to solve signal processing and learning problems globally. Figure 7, for example, plots the recovery error of a randomly initialized run of a subgradient method on a simplified *phase retrieval* problem, an imaging modality underlying X-ray crystallography and which permitted the discovery of the double helix [60, 61]. Given data $\{z_i\}_{i=1}^m = \{(a_i, b_i)\}_{i=1}^m$ contained in $\mathbb{R}^d \times \mathbb{R}$, it seeks $x_\star \in \mathbb{R}^d$ satisfying $(a_i^T x_\star)^2 \approx b_i, (i = 1, \dots, m)$ via minimizing the convex composite *empirical risk*:

$$g(x) := \frac{1}{m} \sum_{i=1}^{m} |(a_i^T x)^2 - b_i| \qquad (\mathcal{PHASE})$$

In Figure 7, I take $d \approx 2^{24}$ and $m = 3d \approx 2^{25}$ and find the whole experiment completed in 30 seconds on a (moderately slow) desktop computer.

How do we understand this favorable behavior of the subgradient method? Let us begin by looking at the behavior near a solution. There, rapid local dynamics tend to arise from good conditioning. Conditioning of a linear system, for example, governs the speed of iterative solvers. It plays a similarly fundamental role in optimization, manifesting as growth: for all x near $\operatorname{argmin} g$, we have

$$g(x) - \inf g \geq \mu \cdot \operatorname{dist}^\alpha(x, \operatorname{argmin} g) \qquad \text{for some } \alpha, \mu > 0, \qquad (\mathcal{GROWTH})$$

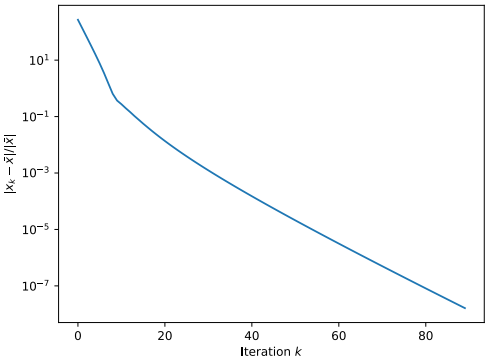


Figure 7: Randomly initialized subgradient method on phase retrieval problem. (Negligible) Confidence intervals omitted.

A positive definite Hessian at a minimum, for example, dictates *quadratic growth* ($\alpha = 2$) and local linear convergence of simple iterative methods. On the other hand, *sharp growth* ($\alpha = 1$, see Figure 8 for illustration) has classically played a central role in subgradient methods, implying local linear convergence for *convex* losses. [62–66]. In the context of Section 3 and the orthogonal decomposition in Figure 4, sharp growth coincides with the case where $f_{\mathcal{U}} = 0$. Because of this, one can expect much faster convergence of the form $f(x_k) - \inf f = O(\exp(-C_f k))$ where C_f depends only on f.

Indeed, in [8, 67], I showed that rapid local dynamics of subgradient methods on sharp losses persist for the convex composite class described in Section 5. Powerful consequences in statistical estimation and learning arise from this result. For illustration, the empirical phase retrieval loss ($\mathcal{PHASE}$) is sharp, a fact first known in the statistical recovery literature [68], where it was interpreted as strong identifiability of the statistical model $x_\star$, rather than as a useful algorithmic device. Recognizing this connection, I showed in [10] that a (properly initialized) subgradient method recovers $x_\star$ with the (nearly) *optimal sample complexity* and the best known *computational complexity* guarantees to date. How deep is this connection between "strong

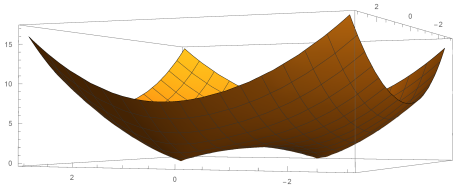


Figure 8: *A sharp loss.*

identifiability" and good conditioning/rapid dynamics? I have found that, beyond phase retrieval, growth is pervasive in statistical estimation, for example, in low-rank matrix recovery where sharp growth takes hold as soon as the number of "measurements" surpasses the information-theoretically minimal (or near minimal) number needed for recovery, leading to similarly strong sample and computational complexity guarantees [11, 12]. Importantly, through nonsmooth (e.g., ℓ_1) penalization techniques, these results open the door to "outlier robust" recovery guarantees with linearly and even quadratically convergent iterative methods (see Figure 9).

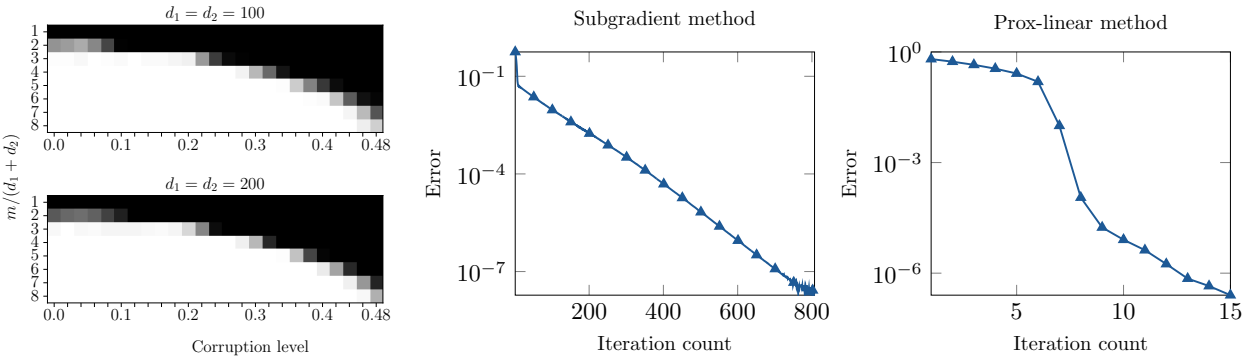
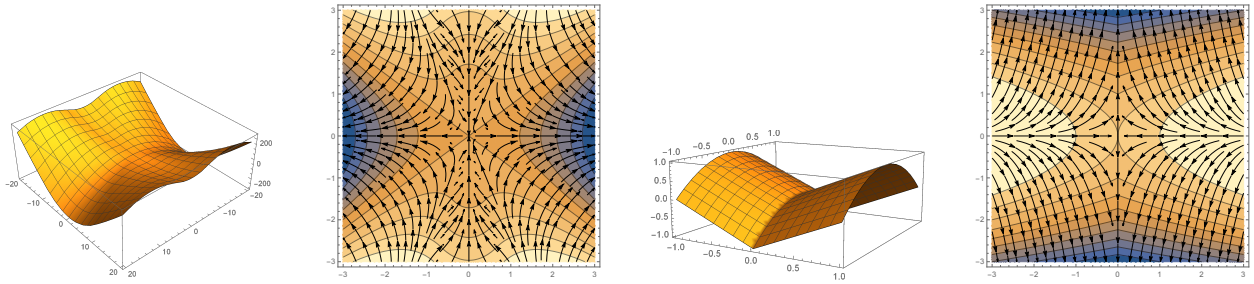


Figure 9: Performance of two iterative methods for a nonsmooth "robust" bilinear sensing problem with outlier corruption [11]. Left: Exact recovery frequencies when varying the percentage of outliers (x-axis) and the amount of "oversampling" $m/(d_1+d_2)$ (y-axis). Center/Right: Recovery error of two iterative methods under 45% corruption by outliers.

7. Avoiding saddle points of nonsmooth function asymptotically. The works discussed thus far studied the behavior of iterative algorithms in two regimes: rapid local convergence near minimizers (Sections 3 and 6) and global asymptotic/finite guarantees for convergence to (possibly nonoptimal) critical points (Sections 4 and 5). While in theory, iterative methods could converge to (locally) suboptimal critical points from random initialization, in practice, this is rare, at least in deep learning and statistical recovery problems (e.g., see Figure 7 where the subgradient method avoids spurious critical points). Thus, a long-standing open problem in nonsmooth optimization is determining whether and when standard training algorithms' limit points tend to local minimizers or saddle points.

For inspiration, I look to the smooth setting, where the seminal papers [26, 27] prove that simple iterative methods (e.g., gradient descent) for C^2 optimization avoid all *strict* saddle points (critical points that have negative curvature) when randomly initialized. If all saddle points are strict, such methods converge to local minima. Later works [69–73] showed that several signal processing and learning problems possessed this *strict-saddle property* and also had no spurious local minimizers, implying that simple randomly initialized methods converge to global minima. In search of a generalization to nonsmooth losses, a tempting conjecture is that negative curvature, suitably defined, implies avoidance of saddle points. Unfortunately, this fails even for simple C^1 functions. For example, Figure 10.a plots a C^1 function such that with constant probability, its randomly initialized gradient flow (Figure 10.b), as well as its discretization to the gradient method, converges to the saddle (the origin). Similar negative results persist even for C^∞ optimization over a single affine inequality [74].



(a) C^1 *loss* f **(b)** *Flow* $\dot{x} = -\nabla f(x)$ **(c)** $f(u, v) = |v| - u^2$ **(d)** *Flow* $\dot{x} \in -\partial f(x)$

Figure 10: *Left: A* C^1 *loss and its flow; Right: A nonsmooth loss with an* active strict saddle *and its flow.*

In the works [13, 14], I resolved this open problem for two classes of iterative algorithms by developing a theory of avoidable nonsmooth saddle points called "active strict saddles." These iterative algorithms include both randomly initialized "proximal" methods [13]—fundamental algorithmic primitives in statistical estimation and learning [75]—and *randomly perturbed* (i.e., noise injected) stochastic subgradient methods (SGD) [14]. Figure 10c plots the prototypical form of an active strict saddle point. Figure 10d illustrates that its randomly initialized *subgradient flow* avoids the origin with probability 1. From the figure, we see that the subgradient flow of f contrasts with that of the pathological example in Figure 10a. Indeed, while both functions have directions of negative curvature, the set of origin-attracted initial conditions of the flow of $-\partial f$ is simply the x-axis—a measure zero set. This favorable behavior of f arises because its nonsmoothness manifests in a structured way: its critical point $\bar{z}$ (the origin) lies on an "active" smooth manifold $\mathcal{M}$ (the u-axis). The function f then varies smoothly along $\mathcal{M}$ (a "partial smoothness" property) and sharply normal to $\mathcal{M}$ meaning:

$$\inf\{\|v\| : v \in \partial f(z),\ z \in U \setminus \mathcal{M}\} > 0, \qquad (\mathcal{ACTIVE})$$

where U is some neighborhood of $\bar{z}$. I call such structured critical points *active strict saddles*, and say a loss $f\colon \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ has the *active strict saddle property* if each of its critical points is either a local minimizer or an active strict saddle. (Note: infinite values of f implicitly impose constraints.) Though the property may appear stringent, it is, in a precise sense, typical: for any tame f satisfying a mild "Clarke regularity" property, I prove the perturbation $f_v(x) = f(x) - \langle v, x\rangle$ has the active strict saddle property for almost all $v \in \mathbb{R}^d$ [13, 14].

This active strict saddle property induces an analogous "orthogonal decomposition" decomposition as in Figure 4. The difference is that the smooth function $f_{\mathcal{U}}$ now has a strict saddle at the critical point rather than a minimizer. In both works [13, 14], the key idea is to leverage classical stable manifold type arguments for the loss $f_{\mathcal{U}}$ or the loss f restricted to $\mathcal{M}$. For example, in [13], I show that proximal methods reach $\mathcal{M}$ in finite time, so smooth dynamics take over, and classical arguments apply. For randomly perturbed subgradient methods, we introduce and verify a property called (a)-regularity[8] that roughly states that $f_{\mathcal{V}}$ is smooth in tangent directions to the manifold up to an error term which is linear in the distance to the manifold; we then use this property to show that the shadow of the iterates of the perturbed subgradient method along the manifold form an approximate gradient descent sequence for f restricted to the manifold. Classical stable manifold arguments then imply nonconvergence to a saddle point [77].

[8] A variant of the Verdier stratification condition [76] in tame geometry.

8. Avoiding saddle points of nonsmooth functions in polynomial time. While asymptotic convergence to local minimizers is desirable, how efficiently do simple iterative algorithms escape active strict saddles and converge to local minima? Here, a negative result surfaces: even for C^2 losses, gradient descent may take exponential time to avoid saddles [78]. Despite failure in general, however, a second line of work proves *randomly perturbed gradient methods* on sufficiently smooth functions avoid saddles in *polynomial time* [79–81]. Is an adaptation to nonsmooth losses possible, and with what complexity? In recent work [15], I developed a promising approach, based on an *inexact gradient method* on the *Moreau smoothing* of f as in ($\mathcal{MOREAU}$). The key insights: for convex composite losses, f and f_γ share all critical points. Moreover, in a neighborhood of an active strict saddle, the Moreau envelope inherits both the smoothness level of f along the active manifold and its negative curvature. Finally, for weakly convex losses, evaluating ∇f_γ *inexactly* amounts to minimizing the *strongly convex problem* in ($\mathcal{MOREAU}$) approximately—a numerically efficient operation, implementable by many algorithms, even the subgradient method ($\mathcal{SM}$) itself. The main conclusion of [82] is thus that a variety of algorithms for nonsmooth optimization can escape strict saddle points of the Moreau envelope at a controlled rate.

9. Asymptotic normality in nonsmooth optimization. Polyak and Juditsky [83] famously showed that the stochastic gradient method for minimizing smooth and strongly convex functions enjoys a central limit theorem: the error between the running average of the iterates and the minimizer, normalized by the square root of the iteration counter, converges to a normal random vector. Moreover, the asymptotic covariance matrix is, in a precise sense "optimal" among any estimation procedure. If an estimate of the covariance is available, e.g., through online methods [84, 85], asymptotic normality guarantees then allow one to construct *confidence intervals* for the iterates of SGD. A long-standing open question is whether similar guarantees – asymptotic normality and optimality – exist for nonsmooth optimization and, more generally, for equilibrium problems. In the work [16], I developed such guarantees under mild conditions that hold both in concrete circumstances (e.g., nonlinear programming) and under "generic" linear perturbations of tame functions.

The guarantees of [16] are already interesting for stochastic nonlinear programming:

$$\min_x \; f(x) = \mathop{\mathbb{E}}_{z\sim\mathcal{P}}[f(x,z)] \qquad \text{subject to} \qquad g_i(x) \le 0 \qquad \text{for } i = 1,\dots,m. \tag{2}$$

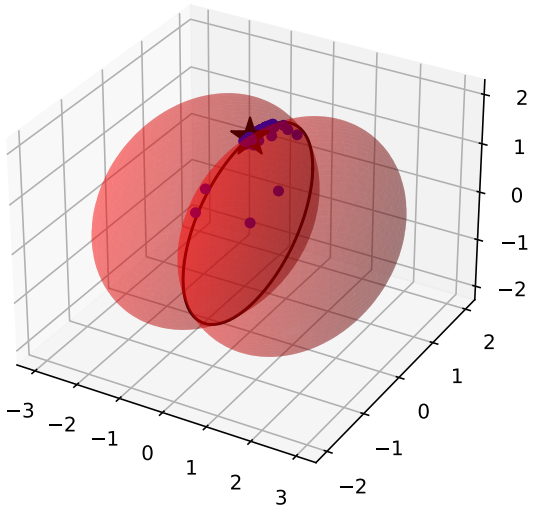
Here, each g_i is a smooth function, and the map $x \mapsto f(x,z)$ is smooth for a.e. $z \sim \mathcal{P}$. Sample average approximation (SAA) and the stochastic proximal-gradient algorithm (SPG) are two standard strategies for solving (2). The former draws a batch of samples $z_1, z_2, \dots, z_k \overset{\text{iid}}{\sim} \mathcal{P}$ and finds a solution x_k to the empirical approximation

$$\min_x \frac{1}{k}\sum_{i=1}^{k} f(x,z_i) \qquad \text{subject to} \qquad g_i(x) \le 0 \qquad \text{for } i = 1,\dots,m. \tag{3}$$

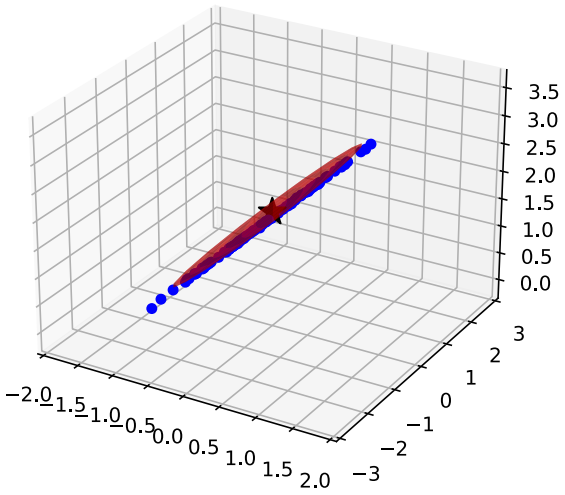
In contrast, the SPG algorithm is a streaming algorithm. At each step it draws a single sample $z_k \sim \mathcal{P}$ in each iteration k and declares the next iterate x_{k+1} to be

$$x_{k+1} \in P_{\mathcal{X}}(x_k - \alpha_k \cdot \nabla f(x_k, z_k)). \tag{4}$$

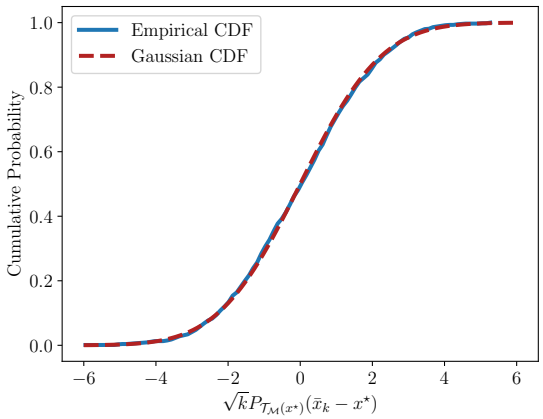
Here, $P_{\mathcal{X}}(\cdot)$ denotes the nearest-point projection onto $\mathcal{X}$. Online algorithms like SPG are usually preferable to SAA since each iteration can be inexpensive, whereas SAA solves the full problem (3). The asymptotic distribution of the SAA estimator is also well-understood [86–88]. In contrast, there



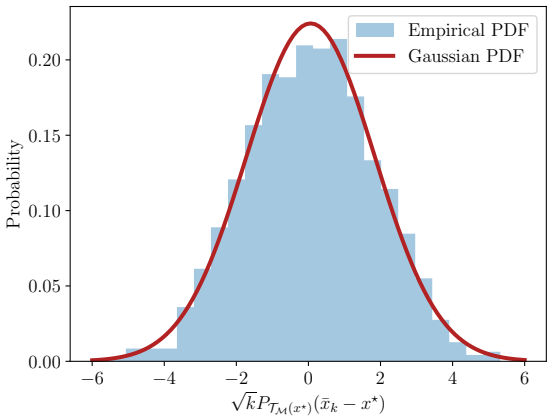
(a) *Feasible region and the iterates* x_k



(b) *The deviations* $\sqrt{k}(\bar{x}_k - x^\star)$ *and the* 95% *confidence region.*



(c) *Empirical vs Gaussian CDF*



(d) *Histogram vs Gaussian density.*

Figure 11: *The stochastic projected gradient method for minimizing* $\mathbb{E}_g[-x_1 + \langle g, x\rangle]$ *over the intersection of two balls centered around* $(-1, 0, 0)$ *and* $(1, 0, 0)$ *of radius two. Here,* $g \sim N(0, I)$*. The optimal solution* $(0, 0, \sqrt{3})$ *(marked with a star) lies on an "active manifold"* $\mathcal{M}$*–the circle depicted in black. The top left figure depicts the iterates generated by a single run of the process initialized at the origin with stepsize* $\eta_k = k^{-3/4}$ *and executed for* 1000 *iterations. The figure on the top right depicts the rescaled deviations* $\sqrt{k}(\bar{x}_k - x^\star)$ *taken over* 100 *runs with* $k = 10^6$*. The two figures clearly show that the iterates rapidly approach the active manifold, and asymptotically the deviations* $\sqrt{k}(\bar{x}_k - x^\star)$ *are supported only along the tangent space to* $\mathcal{M}$ *at* $x^\star$*. The two figures on the second row show the histogram and the empirical CDF, respectively, of the tangent components* $\sqrt{k}P_{T_{\mathcal{M}}(x^\star)}(\bar{x}_k - x^\star)$*, overlaid with the analogous functions for a Gaussian.*

is no known guarantee for the asymptotic performance of the SPG in nonsmooth and constrained settings.

In recent work [16], I prove that under mild assumptions, the running average of the SPG iterates have the same asymptotic distribution as those of SAA. Moreover, both SAA and SPG are asymptotically optimal in a locally minimax sense of Hájek, and Le Cam [89, 90] in both the nonlinear programming problem and a broad family of stochastic equilibrium problems (e.g., games). Specifically, I show that under the three standard conditions—linear independence of

active gradients, strict complementarity, and strong second-order sufficiency—the running average of the SPG iterates $\bar{x}_k = \frac{1}{k}\sum_{i=1}^{k} x_i$ is asymptotically normal and optimal:

$$\sqrt{k}(\bar{x}_k - x^\star) \xrightarrow{D} N\left(0, H^\dagger \cdot \mathrm{Cov}(\nabla f(x^\star, z)) \cdot H^\dagger\right).$$

Here, H admits an explicit description as

$$H = P_{\mathcal{T}} \nabla^2_{xx} \mathcal{L}(x^\star, y^\star) P_{\mathcal{T}} \tag{5}$$

where $\nabla^2_{xx}\mathcal{L}(x^\star, y^\star)$ is the Hessian of the Lagrangian function, the symbol $\dagger$ denotes the Moore-Penrose pseudoinverse, and $P_{\mathcal{T}}$ is the projection onto the subspace "tangent space" $\{\nabla g_i(x^\star)\}^{\perp}_{i\in\mathcal{I}}$ associated to the set of active indices: $\mathcal{I} = \{i : g_i(x^\star) = 0\}$. These three conditions ensure again that the orthogonal decomposition in Figure 4 exists. In this case we have an "active manifold" defined by the active indices $\mathcal{M} := \{x \colon g_i(x) = 0 \ \ (\forall i \in \mathcal{I})\}$ and the identification $\nabla^2_{xx}\mathcal{L}(x^\star, y^\star) = \nabla^2 f_{\mathcal{U}}(x^\star)$.

Figure 11 illustrates the result with the performance of the projected stochastic gradient method for minimizing a linear function over the intersection of two balls. This performance is surprising in light of the work of Duchi and Ruan [91], which uncover a striking gap between the estimation quality of SAA and at least one standard streaming method, called dual averaging [92, 93], for stochastic nonlinear optimization. Indeed, even for the problem of minimizing the expectation of a linear function over a ball, the dual averaging method exhibits a suboptimal asymptotic covariance [91, Section 5.2].[9] In contrast, we see that the stochastic projected gradient method is asymptotically optimal.

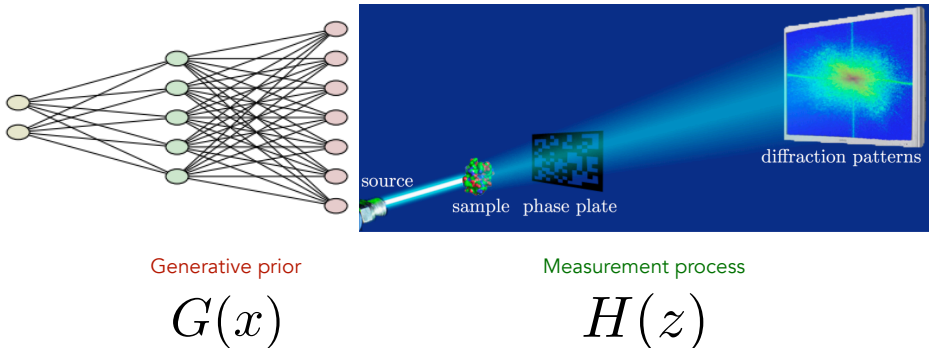
10. An exponential acceleration root-finding methods for "generic" nonsmooth equations. In Section 3, I described my work [4], which develops `NTDescent`, a (nearly) linearly convergent first-order method for "generic" tame optimization problems. `NTDescent` is an exponential improvement on all existing first-order methods for this problem class. Is it possible to improve further, say, to (locally) superlinear convergence of the form $f(x_k) - \inf f = O(2^{-2^k})$? In my recent work [17], I show that this is indeed possible if (i) f is piecewise linear or $f_v(x) = \|F(x) - v\|$ where $F(x) = v$ is a "generic" tame equation; (ii) one knows $\inf f$; and (iii) one solves an (often constant sized) linear system at each iteration. The work also resolves a long-standing open problem for "nonsmooth Newton methods" [94], namely whether there exists a method that solves such problems when F has nonisolated roots and singular "Jacobian" at the root. Finally, the method is (provably) compatible with (nonsmooth) autodifferentiation and can scale to hundreds of thousands of variables, a difficult task for classical Newton-type methods, as demonstrated by my PyTorch library.[10]

Where do nonsmooth equations arise? The work [17] and the associated PyTorch library provide several examples, including neural network training, low-rank matrix sensing, and optimality conditions of nonsmooth optimization problems. In each of these examples, I found that the `SuperPolyak` method improves on gradient methods for the same problem class in terms of time and oracle complexity, even for problems with roughly one million variables. Here, I will consider an interesting though smaller-scale application: imaging with a generative prior.

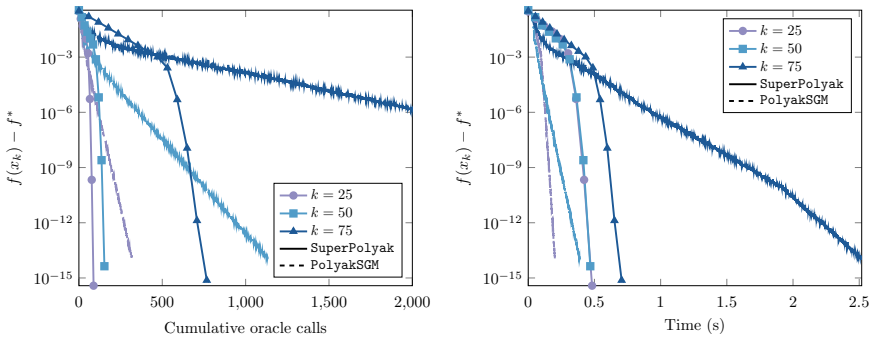
An emerging computational imaging technique is using a pre-trained neural network G as an image prior [95–98]. The prior may help denoise the eventually recovered image or permit one to take fewer measurements/inflict less radiation on biological samples. It is common practice to use a nonsmooth G, e.g., one with ReLU activations. Given G and supposing the imaging task has a

[9] In contrast, in the particular case that $\mathcal{X}$ is polyhedral and convex, the dual averaging method is optimal [91].
[10] PyTorch code available at: `https://github.com/COR-OPT/SuperPolyak.py`.

"forward model" H that generates "measurements" $b = H(z_\star)$ of an image $z_\star$, the mathematical goal is to recover $z_\star$ by solving the nonsmooth equation $H(G(x)) = b$.[11] Figure 12 illustrates the performance of the method `SuperPolyak` on the objective function $f(x) = \|H(G(x)) - b\|$, where in this case, H is a linear mapping and G is a random ReLU network. The plot shows `SuperPolyak` outperforms a classical first-order method (`PolyakSGM`) in terms of time and oracle complexity.



(a) *Generative prior used in an imaging pipeline. Figure on right borrowed from [99].*



(b) *SuperPolyak method of [17] on a compressive sensing problem with a generative prior. Here, each oracle call of* `PolyakSGM` *and* `SuperPolyak` *have the same cost: a subgradient of the loss* $\|H(G(x)) - b\|$ *where the latent code* x *is* k*-dimensional.*

Figure 12: *Solving a compressive sensing problem with* `SuperPolyak`.

How and why does `SuperPolyak` work? The principle behind superpolyak is similar to that underlying the classical Newton method for smooth equations, which linearizes the equation $F(x) = 0$ and solves the linearization. Two issues with this classical approach are that f is now nonsmooth, so Taylor expansions are no longer available, and $f\colon \mathbb{R}^d \to \mathbb{R}$ is no longer a "mapping" but a function. Despite the nonexistence of a Taylor expansion, Lipschitz tame functions are known to satisfy a related "partial smoothness property" called "semismoothness" [100]. The property states that for any root $\bar{x}$ of f, the approximate holds:

$$f(x) + \langle v, \bar{x} - x \rangle = o(\|\bar{x} - x\|) \qquad \text{as } x \to \bar{x} \text{ and } v \in \partial f(x).$$

Roughly speaking, semismoothness provides an equation that the root of f nearly satisfies, though there are infinitely many such solutions to the equation. Thus, a natural strategy is to find the

[11] I ignore the noise issue to keep the discussion simple.

closest root of this equation. This approach gives rise to `PolyakSGM` illustrated in Figure 12b. Unfortunately, as shown in the plot, `PolyakSGM` only converges linearly, not superlinearly. To "repair" the convergence of Newton's method, the strategy of [4] is to choose multiple base points at which to linearize and then solve all linearizations simultaneously. Specifically, the `SuperPolyak` method constructs a sequence of iterates $x_k \in \mathbb{R}^d$. At the kth iteration, it successively solves a sequence of least squares problems arising from "linearizing" the function f at certain "bundle points" y_i: solve

$$y_j := \operatorname*{argmin}_{x \in \mathbb{R}^d} \|x - x_k\|^2 \qquad \text{subject to: } \left[f(y_i) + \langle v_i, x - y_i\rangle\right]_{i=0}^{j-1} = 0 \tag{6}$$

and choose $v_j \in \partial f(y_j)$ arbitrarily for $i = 1, \ldots$. Thus, each time one solves such a collection of linearizations, one adds the solution of the linearization back into the bundle and re-solves the system. While this process could go indefinitely, I prove that (locally) that after at most d steps, we find a point that superlinearly improves on x_k:

$$x_{k+1} \in \operatorname*{argmin}_{y \in \{y_i\}_{i \le d}} f(y) \qquad \text{satisfies} \qquad f(x_{k+1}) = o(f(x_k)) \quad \text{as } k \to \infty.$$

Moreover, we often find a bundle point y_i that superlinearly improves on x_k when $i \ll d$, so we need only solve a constant-sized linear system! In addition, although the naive linear algebra cost of finding $y_1, \ldots y_d$ is $O(d^4)$, I show that one can incrementally build up this bundle $O(d^3)$ arithmetic operations.

11. When spectral updates help in deep learning.

In recent work [2], I studied when and why *spectral gradient methods* in deep learning—notably `Muon` [3]—are expected to outperform standard Euclidean updates. In 2024, these methods started to rapidly gain traction because they can match or surpass `Adam` in NanoGPT-scale runs, including in the modded-NanoGPT "speedrunning" ecosystem [101, 102], and in larger-scale pretraining studies [103]. Here, "spectral" has a concrete meaning: one performs steepest descent in a non-Euclidean matrix norm, namely the spectral norm (Schatten-∞ geometry), whose dual norm is the nuclear norm. This viewpoint already appeared in 2015 in the work of Carlson et al. on PSD/SSD and related methods [104, 105], though it did not substantially permeate mainstream deep-learning practice until much more recently. Beyond these community-scale benchmarks, it appears that Muon-style updates are now being explored in much larger runs as well (e.g., Kimi K2 reports Muon in its training runs [106]). This raises a basic question: *when should one expect a spectral step to yield a larger decrease in the loss than a Euclidean gradient step?*

To answer this question, we first discuss what a spectral update is at a high level. Many parameters in modern networks are *weight matrices* (or blocks that can be reshaped into matrices). If $W \in \mathbb{R}^{m \times k}$ is such a block and $G = \nabla_W \mathcal{L}(W)$ is its gradient, spectral methods replace the raw gradient direction by (an approximation of) its *polar factor*. Writing the singular value decomposition $G = U\Sigma V^\top$, the polar factor is

$$\operatorname{polar}(G) := UV^\top,$$

i.e., it keeps the singular vectors and discards the singular values. In the simplest form of spectral gradient descent (`SpecGD` [107]) one uses the update

$$W^+ \;=\; W \;-\; \alpha \cdot \|G\|_* \cdot \operatorname{polar}(G), \tag{7}$$

where $\|\cdot\|_*$ is the nuclear norm. `Muon` [3] is a practical variant: it first forms an exponential moving average M_t of the gradients and then applies an approximate polar-factor map to M_t (implemented

via the Newton–Schulz algorithm, which estimates polar$(\cdot)$ using only a handful of matrix–matrix products).

In [2], I observe the following: for a given parameter block, when an inequality comparing a (type of) "effective rank" of the gradient to the "stable rank" of the incoming activations holds, one can expect improved performance of `SpecGD` over `GD`. To understand the inequality, consider first the simple, but representative, random feature regression model

$$\min_{W\in\mathbb{R}^{m\times k}} \ \mathcal{L}(W) := \tfrac{1}{2n}\|WA - Y\|_F^2, \tag{8}$$

where A should be viewed as the *post-activation matrix* produced by the upstream layers and W is the matrix of the current layer. Since (8) is quadratic, a Taylor expansion gives

$$\mathcal{L}(W+U) = \mathcal{L}(W) + \langle \nabla\mathcal{L}(W), U\rangle + \tfrac{1}{2n}\|UA\|_F^2.$$

The last term controls the local curvature in directions U, and it admits two natural upper bounds:

$$\tfrac{1}{n}\|UA\|_F^2 \le \tfrac{1}{n}\|A\|_{\mathrm{op}}^2 \cdot \|U\|_F^2 \qquad \text{and} \qquad \tfrac{1}{n}\|UA\|_F^2 \le \tfrac{1}{n}\|A\|_F^2 \cdot \|U\|_{\mathrm{op}}^2.$$

Minimizing the corresponding upper quadratic models of the loss induced by these bounds yields, respectively, a Euclidean gradient step and a spectral step. The resulting *guaranteed* one-step decrease in the loss scales like

$$\Delta_{\mathrm{GD}} \asymp \frac{\|G\|_F^2}{\|A\|_{\mathrm{op}}^2} \qquad \text{and} \qquad \Delta_{\mathrm{Spec}} \asymp \frac{\|G\|_*^2}{\|A\|_F^2},$$

so spectral descent is favored whenever

$$\underbrace{\frac{\|G\|_*^2}{\|G\|_F^2}}_{=:\ \mathrm{nr}(G)} \ \ge \ \underbrace{\frac{\|A\|_F^2}{\|A\|_{\mathrm{op}}^2}}_{=:\ \mathrm{st}(A)}. \tag{9}$$

Here $\mathrm{st}(A)$ is the *stable rank* of the incoming features and $\mathrm{nr}(G)$ is also a type of effective rank of the gradient, which we call the *nuclear rank*.[12] Although (9) is derived in a quadratic toy model, the same comparison extends (with appropriate bookkeeping) to layered MLPs and transformer blocks: for each matrix block W_ℓ multiplying an activation matrix $A_{\ell-1}$, one expects a spectral update to be most effective when $\mathrm{nr}(G_\ell)$ dominates $\mathrm{st}(A_{\ell-1})$.

To use (9), one needs to understand when the stable rank of post-activations is small and the nuclear rank of the corresponding gradients is large. In the work [2], I identify two prevalent sources of low-stable-rank activations in deep learning:

1. *Mean spikes induced by activations.* For many common nonlinearities (including ReLU and squared-ReLU), Gaussian inputs induce a mean spike: if $\gamma \sim \mathcal{N}(0,1)$ and $m_1 := \mathbb{E}[\sigma(\gamma)] \neq 0$ while $m_2 := \mathbb{E}[\sigma(\gamma)^2] < \infty$, then the post-activation matrices $\sigma(WX)$ at Gaussian initialization exhibit a dominant rank-one component, forcing $\mathrm{st}(\sigma(WX))$ to be bounded by a numerical constant controlled by m_2/m_1^2 [2].

2. *Token-indicator structure in transformers.* Let $H \in \mathbb{R}^{V\times n}$ be the token-indicator matrix for a length-n sequence (the tth column is the one-hot vector for token i_t). Writing $p_{\max}$ for the empirical frequency of the most common token, a direct calculation gives $\mathrm{st}(H) = 1/p_{\max}$ [2, Section 2]. Token frequencies in natural language corpora are heavy-tailed (Zipf-type behavior), so $p_{\max}$ is not extremely small and $\mathrm{st}(H)$ is modest.

[12] For any matrix B, we have $1 \le \mathrm{st}(B) \le \mathrm{nr}(B) \le \mathrm{rank}(B)$, and $\mathrm{nr}(B)$ can be much larger than $\mathrm{st}(B)$.

Block	Stable rank	Assumptions
Linear map	$\mathrm{st}(WX) \lesssim \mathrm{st}(X)$	—
Pointwise nonlinearity	$\mathrm{st}(\sigma(WX)) \lesssim \mathrm{st}(X)$	$\lvert\mathbb{E}\sigma'(g)\rvert > 0$
Residual connection	$\mathrm{st}(X+WH) \lesssim \min\{\mathrm{st}(X), \mathrm{st}(H)\}$	$\lVert X\rVert_F \asymp \lVert H\rVert_F$
RMSNorm	$\mathrm{st}\left(\left[\frac{x_1}{\lVert x_1\rVert_2} \quad \cdots \quad \frac{x_n}{\lVert x_n\rVert_2}\right]\right) \lesssim \mathrm{st}(X)$	nearly const. column norms
Gating	$\mathrm{st}(\sigma(VZ) \odot WX) \lesssim \mathrm{st}(X)$	nearly const. column norms, $\lvert\mathbb{E}\sigma(g)\rvert > 0$

Table 1: Stable-rank propagation rules for common network operations at Gaussian initialization; see [2] for precise constants and proofs.

To pass from these base cases to full architectures—especially transformers, where each block contains normalization, attention mixing, residual updates, and (possibly gated) MLPs—I prove that the stable rank is preserved up to constant factors under the standard building blocks used in modern networks; see [2, Section 2]. The table below summarizes the form of these bounds at Gaussian initialization.

Empirically, this low-stable-rank structure persists well into training. For example, Figure 13 tracks the stable rank of MLP post-activations in a modded NanoGPT run and shows that, despite a maximal possible ambient rank in the thousands, the observed stable ranks remain close to a small numerical constant.

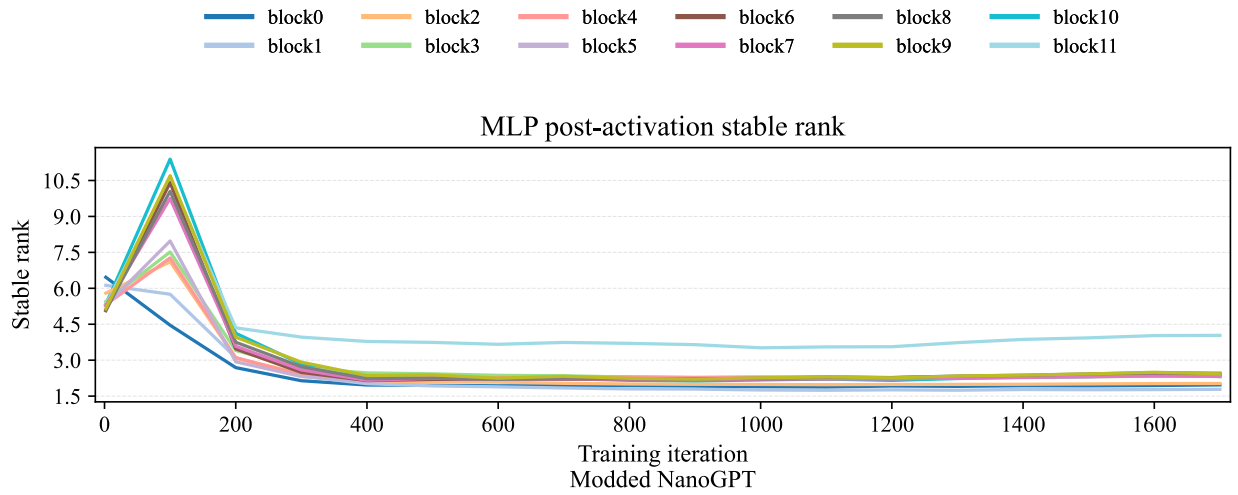


Figure 13: *Stable rank of MLP post-activations in a modded NanoGPT run (see [101, 102]). The maximal possible rank here is* 3072*, yet the observed stable rank remains far below this ceiling throughout training.*

The other side of (9) is the gradient nuclear rank. In [2] I prove this quantity becomes large in a spiked random feature regression model: after a short burn-in, the Euclidean gradient's nuclear rank grows with the ambient dimension and remains large over a window of time steps that scale linearly with the dimension. Even in the simple random-feature experiment in Figure 14, one sees (9) activated along the trajectory and the resulting advantage of `SpecGD` over `GD`.

In transformer training, the same picture appears in the modded NanoGPT setting where Muon's effectiveness was first demonstrated: intermediate blocks often have low-stable-rank incoming activations (Figure 13) while the corresponding gradients maintain large nuclear-to-Frobenius

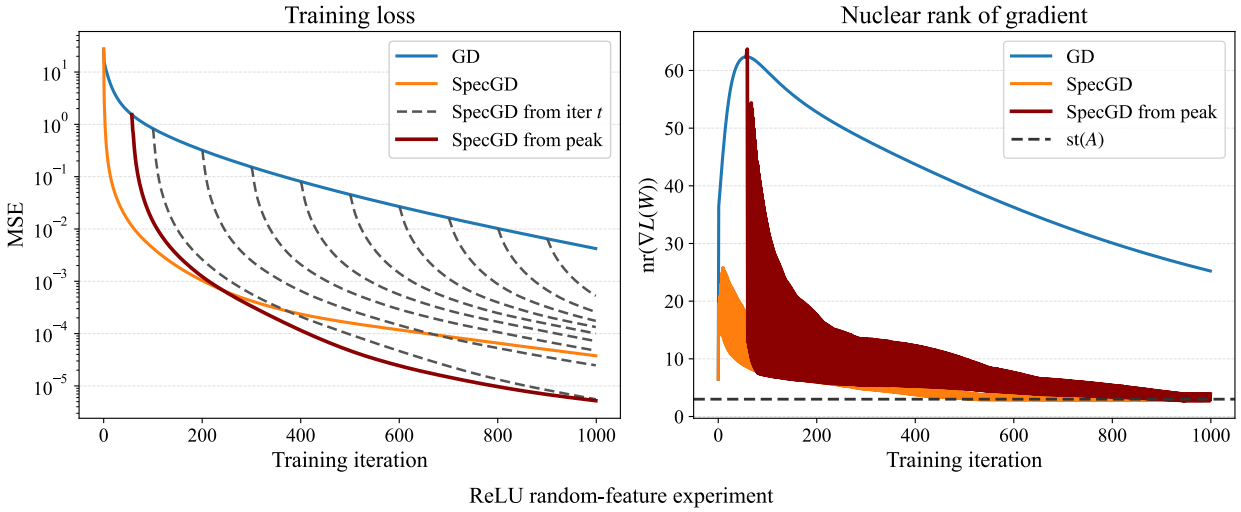


Figure 14: *Random feature regression illustration from [2]. We compare* `GD` *and* `SpecGD` *on* $\min_W \mathcal{L}(W) = \frac{1}{2n}\|WA - Y\|_F^2$ *with* $W \in \mathbb{R}^{100\times 100}$, $A = \sigma(W_1 X)$, $W_1 \in \mathbb{R}^{100\times 50}$, $X \in \mathbb{R}^{100\times 400}$, *and* $\sigma(t) = \max\{t, 0\}$, *initialized at* 0. *Left: suboptimality gap along* `GD` *(blue) and* `SpecGD` *(gold); dashed curves show the gap if we restart* `SpecGD` *from the current* `GD` *iterate every 100 steps. Right: gradient nuclear rank* $\mathrm{nr}(\nabla\mathcal{L}(W))$ *along the trajectories, with the dashed black level* $\mathrm{st}(A)$. *The condition* $\mathrm{nr}(G) \geq \mathrm{st}(A)$ *holds along the trajectory and predicts the observed speedup.*

ratios. Figure 15, for instance, reports gradient nuclear ranks for NanoGPT MLP blocks.

Putting these observations together, (9) thus offers an explanation for why spectral updates help in deep learning. Low-stable-rank post-activations induce ill-conditioning, while spectral updates are less sensitive to ill-conditioning and more effective when the corresponding gradient blocks have high nuclear rank. In practice, this suggests a simple rule of thumb: if a layer's incoming features have small stable rank and the corresponding gradient block has large nuclear rank, that block is a good candidate for `Muon`/`SpecGD`; if the nuclear rank is small, it can be better to leave the block on Euclidean-style updates (as is often done in practice for embeddings and classifier heads; see [3]).

Looking forward, the most interesting conceptual issue to me is that this degeneracy seems *inherent*: low-stable-rank representations arise both from activation statistics (mean spikes) and from discrete token structure, and stable-rank propagation suggests they persist through composition. A major open problem is therefore to design *implementable* training methods that are even less sensitive to this ill-conditioning than spectral gradient methods, while remaining scalable.

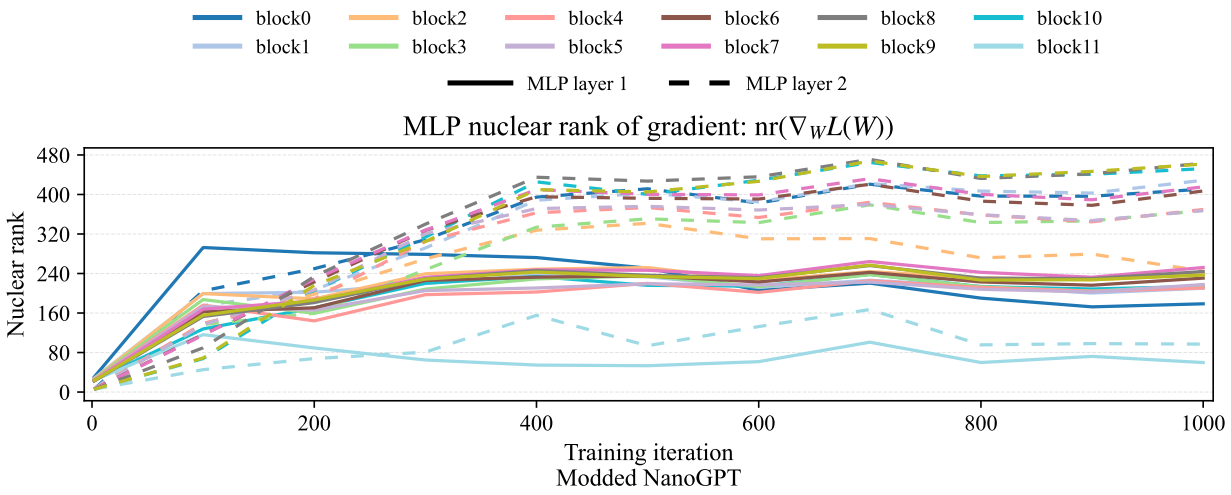


Figure 15: *Gradient nuclear ranks for MLP blocks in a modded NanoGPT run [2, 101]. In the regime where these ratios are large while the incoming activations have small stable rank (Figure 13), the comparison (9) predicts that* `SpecGD`*-style updates are particularly advantageous.*

12. References Cited.

[1] A. Ioffe. "An invitation to tame optimization". In: *SIAM Journal on Optimization* 19.4 (2009), pp. 1894–1917.

[2] D. Davis and D. Drusvyatskiy. "When do spectral gradient updates help in deep learning?" In: *arXiv preprint arXiv:2512.04299* (2025).

[3] K. Jordan, Y. Jin, V. Boza, J. You, F. Cesista, L. Newhouse, and J. Bernstein. *Muon: An optimizer for hidden layers in neural networks*. 2024.

[4] D. Davis and L. Jiang. "A Local Nearly Linearly Convergent First-Order Method for Nonsmooth Functions with Quadratic Growth". In: *Foundations of Computational Mathematics* 25.3 (2025), pp. 943–1024.

[5] D. Davis, D. Drusvyatskiy, S. Kakade, and J. D. Lee. "Stochastic Subgradient Method Converges on Tame Functions". In: *Foundations of Computational Mathematics* 20.1 (2020), pp. 119–154.

[6] D. Davis and D. Drusvyatskiy. "Stochastic Model-Based Minimization of Weakly Convex Functions". In: *SIAM Journal on Optimization* 29.1 (2019), pp. 207–239.

[7] D. Davis and B. Grimmer. "Proximally guided stochastic subgradient method for nonsmooth, nonconvex problems". In: *SIAM Journal on Optimization* 29.3 (2019), pp. 1908–1930.

[8] D. Davis, D. Drusvyatskiy, K. J. MacPhee, and C. Paquette. "Subgradient Methods for Sharp Weakly Convex Functions". In: *Journal of Optimization Theory and Applications* 179.3 (2018), pp. 962–982.

[9] D. Davis, D. Drusvyatskiy, and V. Charisopoulos. "Stochastic algorithms with geometric step decay converge linearly on sharp functions". In: *Mathematical Programming* 207.1-2 (2023), pp. 145–190. DOI: 10.1007/s10107-023-02003-w.

[10] D. Davis, D. Drusvyatskiy, and C. Paquette. "The nonsmooth landscape of phase retrieval". In: *IMA Journal of Numerical Analysis* 40.4 (Jan. 2020), pp. 2652–2695.

[11] V. Charisopoulos, D. Davis, M. Díaz, and D. Drusvyatskiy. "Composite optimization for robust rank one bilinear sensing". In: *Information and Inference: A Journal of the IMA* (Oct. 2020). iaaa027. DOI: 10.1093/imaiai/iaaa027.

[12] V. Charisopoulos, Y. Chen, D. Davis, M. Díaz, L. Ding, and D. Drusvyatskiy. "Low-Rank Matrix Recovery with Composite Optimization: Good Conditioning and Rapid Convergence". In: *Foundations of Computational Mathematics* (2021). DOI: 10.1007/s10208-020-09490-9.

[13] D. Davis and D. Drusvyatskiy. "Proximal Methods Avoid Active Strict Saddles of Weakly Convex Functions". In: *Foundations of Computational Mathematics* (2021). DOI: 10.1007/s10208-021-09516-w.

[14] D. Davis, D. Drusvyatskiy, and L. Jiang. "Active manifolds, stratifications, and convergence to local minima in nonsmooth optimization". In: *Foundations of Computational Mathematics* (2025). DOI: 10.1007/s10208-025-09691-0.

[15] D. Davis, M. Díaz, and D. Drusvyatskiy. "Escaping Strict Saddle Points of the Moreau Envelope in Nonsmooth Optimization". In: *SIAM Journal on Optimization* 32.3 (2022), pp. 1958–1983. DOI: 10.1137/21M1430868. eprint: https://doi.org/10.1137/21M1430868.

[16] D. Davis, D. Drusvyatskiy, and L. Jiang. "Asymptotic normality and optimality in nonsmooth stochastic approximation". In: *The Annals of Statistics* 52.4 (2024), pp. 1485–1508. DOI: 10.1214/24-AOS2401.

[17] V. Charisopoulos and D. Davis. "A superlinearly convergent subgradient method for sharp semismooth problems". In: *Mathematics of Operations Research* 49.3 (2024), pp. 1678–1709.

[18] D. Donoho. "Data Science at the Singularity". In: *arXiv preprint arXiv:2310.00865* (2023).

[19] X. Lu, M. Pham, E. Negrini, D. Davis, S. J. Osher, and J. Miao. "Computational Microscopy beyond Perfect Lenses". In: *arXiv preprint arXiv:2306.11283* (2023).

[20] A. V. Sadovsky, D. Davis, and D. R. Isaacson. "Separation-compliant, optimal routing and control of scheduled arrivals in a terminal airspace". In: *Transportation Research Part C: Emerging Technologies* 37 (2013), pp. 157–176.

[21] A. V. Sadovsky, D. Davis, and D. R. Isaacson. "Efficient computation of separation-compliant speed advisories for air traffic arriving in terminal airspace". In: *Journal of Dynamic Systems, Measurement, and Control* 136.4 (2014), p. 041027.

[22] D. R. Isaacson, A. V. Sadovsky, and D. Davis. "Tactical scheduling for precision air traffic operations: Past research and current problems". In: *Journal of Aerospace Information Systems* 11.4 (2014), pp. 234–257.

[23] R. Kumar, O. López, D. Davis, A. Y. Aravkin, and F. J. Herrmann. "Beating Level-Set Methods for 5-D Seismic Data Interpolation: A Primal-Dual Alternating Approach". In: *IEEE Transactions on Computational Imaging* 3.2 (2017), pp. 264–274. DOI: 10.1109/TCI.2017.2693966.

[24] J. Dong, N. Karianakis, D. Davis, J. Hernandez, J. Balzer, and S. Soatto. "Multi-view feature engineering and learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3251–3260.

[25] D. Davis, J. Balzer, and S. Soatto. "Asymmetric sparse kernel approximations for large-scale visual search". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 2107–2114.

[26] J. Lee, M. Simchowitz, M. Jordan, and B. Recht. "Gradient descent only converges to minimizers". In: *Conference on learning theory*. 2016, pp. 1246–1257.

[27] J. Lee, I. Panageas, G. Piliouras, M. Simchowitz, M. Jordan, and B. Recht. "First-order Methods Almost Always Avoid Strict Saddle Points". In: *Math. Program.* 176.1-2 (July 2019), pp. 311–337. DOI: 10.1007/s10107-019-01374-3.

[28] D. Drusvyatskiy, A. D. Ioffe, and A. S. Lewis. "Generic minimizing behavior in semialgebraic optimization". In: *SIAM Journal on Optimization* 26.1 (2016), pp. 513–534.

[29] A. S. Nemirovskij and D. B. Yudin. "Problem complexity and method efficiency in optimization". In: (1983).

[30] Y. Nesterov. *Introductory lectures on convex optimization*. Vol. 87. Applied Optimization. A basic course. Kluwer Academic Publishers, Boston, MA, 2004, pp. xviii+236. DOI: 10.1007/978-1-4419-8853-9.

[31] S. Bubeck et al. "Convex optimization: Algorithms and complexity". In: *Foundations and Trends® in Machine Learning* 8.3-4 (2015), pp. 231–357.

[32] B. T. Polyak. "Minimization of unsmooth functionals". In: *USSR Computational Mathematics and Mathematical Physics* 9.3 (1969), pp. 14–29. DOI: 10.1016/0041-5553(69)90061-5.

[33] S. J. Wright. "Identifiable surfaces in constrained optimization". In: *SIAM Journal on Control and Optimization* 31.4 (1993), pp. 1063–1079.

[34] C. Lemaréchal, F. Oustry, and C. Sagastizábal. "The $\mathcal{U}$-Lagrangian of a convex function". In: *Transactions of the American mathematical Society* 352.2 (2000), pp. 711–729.

[35] R. Mifflin and C. Sagastizábal. "A $\mathcal{VU}$-algorithm for convex minimization". In: *Mathematical Programming* 104.2 (2005), pp. 583–608. DOI: 10.1007/s10107-005-0630-3.

[36] A. S. Lewis. "Active sets, nonsmoothness, and sensitivity". In: *SIAM Journal on Optimization* 13.3 (2002), pp. 702–725.

[37] D. Drusvyatskiy and A. S. Lewis. "Optimality, identifiability, and sensitivity". In: *Mathematical Programming* 147.1 (2014). Citations refer to long version arXiv:1207.6628, pp. 467–498.

[38] F. Clarke. *Optimization and Nonsmooth Analysis*. Wiley Interscience, New York, 1983.

[39] J. Bolte and E. Pauwels. "Conservative set valued fields, automatic differentiation, stochastic gradient methods and deep learning". In: *Mathematical Programming* 188.1 (2021), pp. 19–51.

[40] J.-J. Moreau. "Proximité et dualité dans un espace hilbertien". In: *Bulletin de la Société mathématique de France* 93 (1965), pp. 273–299.

[41] C. Daskalakis, D. J. Foster, and N. Golowich. "Independent policy gradient methods for competitive reinforcement learning". In: *Advances in neural information processing systems* 33 (2020), pp. 5527–5540.

[42] S. Leonardos, W. Overman, I. Panageas, and G. Piliouras. "Global convergence of multi-agent policy gradient in markov potential games". In: *arXiv preprint arXiv:2106.01969* (2021).

[43] Q. Wang, C. P. Ho, and M. Petrik. "On the convergence of policy gradient in robust mdps". In: *arXiv preprint arXiv:2212.10439* (2022).

[44] S. Nietert, Z. Goldfeld, R. Sadhu, and K. Kato. "Statistical, robustness, and computational guarantees for sliced wasserstein distances". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 28179–28193.

[45] D. Drusvyatskiy and L. Xiao. "Stochastic optimization with decision-dependent distributions". In: *Mathematics of Operations Research* 48.2 (2023), pp. 954–998.

[46] D. Zhu, G. Li, B. Wang, X. Wu, and T. Yang. "When auc meets dro: Optimizing partial auc for deep learning with non-convex convergence guarantee". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 27548–27573.

[47] Y. Cheng, I. Diakonikolas, R. Ge, and M. Soltanolkotabi. "High-dimensional robust mean estimation via gradient descent". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1768–1778.

[48] Y. Cheng, I. Diakonikolas, R. Ge, S. Gupta, D. Kane, and M. Soltanolkotabi. "Outlier-robust sparse estimation via non-convex optimization". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 7318–7327.

[49] L. Zhu, M. Gürbüzbalaban, and A. Ruszczyński. "Distributionally robust learning with weakly convex losses: Convergence rates and finite-sample guarantees". In: *arXiv preprint arXiv:2301.06619* (2023).

[50] D. Davis, D. Drusvyatskiy, and Z. Shi. "Stochastic optimization over proximally smooth sets". In: *SIAM Journal on Optimization* 35.1 (2025), pp. 157–179.

[51] X. Li, S. Chen, Z. Deng, Q. Qu, Z. Zhu, and A. Man-Cho So. "Weakly convex optimization over Stiefel manifold using Riemannian subgradient-type methods". In: *SIAM Journal on Optimization* 31.3 (2021), pp. 1605–1634.

[52] Z. Chen, S. T. Maguluri, S. Shakkottai, and K. Shanmugam. "Finite-sample analysis of contractive stochastic approximation using smooth convex envelopes". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 8223–8234.

[53] X. Li, Z. Zhu, A. M.-C. So, and J. D. Lee. "Incremental methods for weakly convex optimization". In: *arXiv preprint arXiv:1907.11687* (2019).

[54] Y. Hu, S. Zhang, X. Chen, and N. He. "Biased stochastic gradient descent for conditional stochastic optimization". In: *arXiv preprint arXiv:2002.10790* (2020).

[55] D. Davis, D. Drusvyatskiy, and K. J. MacPhee. "Stochastic model-based minimization under high-order growth". In: *arXiv preprint arXiv:1807.00255* (2018).

[56] M. Liu, H. Rafique, Q. Lin, and T. Yang. "First-order convergence theory for weakly-convex-weakly-concave min-max problems". In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 7651–7684.

[57] K. K. Thekumparampil, P. Jain, P. Netrapalli, and S. Oh. "Efficient algorithms for smooth minimax optimization". In: *Advances in Neural Information Processing Systems* 32 (2019).

[58] T. Lin, C. Jin, and M. I. Jordan. "Near-optimal algorithms for minimax optimization". In: *Conference on Learning Theory*. PMLR. 2020, pp. 2738–2779.

[59] T. Lin, C. Jin, and M. Jordan. "On gradient descent ascent for nonconvex-concave minimax problems". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 6083–6093.

[60] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev. "Phase retrieval with application to optical imaging: a contemporary overview". In: *IEEE signal processing magazine* 32.3 (2015), pp. 87–109.

[61] J. D. Watson, F. H. Crick, et al. "A structure for deoxyribose nucleic acid". In: *Nature* 171.4356 (1953), pp. 737–738.

[62] J. Goffin. "On convergence rates of subgradient optimization methods". In: *Math. Program.* 13.3 (1977), pp. 329–347.

[63] I. Eremin. "The relaxation method of solving systems of inequalities with convex functions on the left-hand side". In: *Dokl. Akad. Nauk SSSR* 160 (1965), pp. 994–996.

[64] B. Poljak. "Minimization of Unsmooth Functionals". In: *USSR Computational Mathematics and Mathematical Physics* 9 (1969), pp. 14–29.

[65] N. Shor. "The rate of convergence of the method of the generalized gradient descent with expansion of space". In: *Kibernetika (Kiev)* 2 (1970), pp. 80–85.

[66] B. Poljak. "Subgradient methods: a survey of Soviet research". In: *Nonsmooth optimization (Proc. IIASA Workshop, Laxenburg, 1977)*. Vol. 3. IIASA Proc. Ser. Pergamon, Oxford-New York, 1978, pp. 5–29.

[67] D. Davis, D. Drusvyatskiy, and V. Charisopoulos. "Stochastic algorithms with geometric step decay converge linearly on sharp functions". In: *Mathematical Programming* 207.1-2 (2023), pp. 145–190. DOI: 10.1007/s10107-023-02003-w.

[68] Y. Eldar and S. Mendelson. "Phase retrieval: stability and recovery guarantees". In: *Appl. Comput. Harmon. Anal.* 36.3 (2014), pp. 473–494. DOI: 10.1016/j.acha.2013.08.003.

[69] R. Ge, J. Lee, and T. Ma. "Matrix completion has no spurious local minimum". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2973–2981.

[70] J. Sun, Q. Qu, and J. Wright. "When are nonconvex problems not scary?" In: *arXiv preprint arXiv:1510.06096* (2015).

[71] S. Bhojanapalli, B. Neyshabur, and N. Srebro. "Global Optimality of Local Search for Low Rank Matrix Recovery". In: *arXiv preprint arXiv:1605.07221* (2016). https://arxiv.org/abs/1605.07221.

[72] R. Ge, C. Jin, and Y. Zheng. "No spurious local minima in nonconvex low rank problems: A unified geometric analysis". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1233–1242.

[73] J. Sun, Q. Qu, and J. Wright. "A geometric analysis of phase retrieval". In: *Foundations of Computational Mathematics* 18.5 (2018), pp. 1131–1198.

[74] M. Nouiehed, J. D. Lee, and M. Razaviyayn. "Convergence to second-order stationarity for constrained non-convex optimization". In: *arXiv preprint arXiv:1810.02024* (2018).

[75] N. Parikh and S. Boyd. "Proximal Algorithms". In: *Found. Trends Optim.* 1.3 (Jan. 2014), pp. 127–239. DOI: 10.1561/2400000003.

[76] J.-L. Verdier. "Stratifications de Whitney et théoreme de Bertini-Sard". In: *Inventiones mathematicae* 36.1 (1976), pp. 295–312.

[77] R. Pemantle. "Nonconvergence to unstable points in urn models and stochastic approximations". In: *The Annals of Probability* 18.2 (1990), pp. 698–712.

[78] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai. "Gradient descent finds global minima of deep neural networks". In: *International Conference on Machine Learning*. 2019, pp. 1675–1685.

[79] S. Du, C. Jin, J. Lee, M. Jordan, A. Singh, and B. Poczos. "Gradient descent can take exponential time to escape saddle points". In: *Advances in neural information processing systems*. 2017, pp. 1067–1077.

[80] R. Jin C.and Ge, P. Netrapalli, S. Kakade, and M. Jordan. "How to escape saddle points efficiently". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1724–1732.

[81] R. Ge, F. Huang, C. Jin, and Y. Yuan. "Escaping from saddle points—online stochastic gradient for tensor decomposition". In: *Conference on Learning Theory*. 2015, pp. 797–842.

[82] D. Davis, M. Díaz, and D. Drusvyatskiy. "Escaping strict saddle points of the Moreau envelope in nonsmooth optimization". In: *SIAM Journal on Optimization* 32.3 (2022), pp. 1958–1983. DOI: 10.1137/21M1430868.

[83] B. T. Polyak and A. B. Juditsky. "Acceleration of stochastic approximation by averaging". In: *SIAM journal on control and optimization* 30.4 (1992), pp. 838–855.

[84] A. Roy and K. Balasubramanian. "Online covariance estimation for stochastic gradient descent under Markovian sampling". In: *arXiv preprint arXiv:2308.01481* (2023).

[85] W. Zhu, X. Chen, and W. B. Wu. "Online covariance matrix estimation in stochastic gradient descent". In: *Journal of the American Statistical Association* 118.541 (2023), pp. 393–404.

[86] A. J. King and R. T. Rockafellar. "Asymptotic theory for solutions in statistical estimation and stochastic programming". In: *Mathematics of Operations Research* 18.1 (1993), pp. 148–162.

[87] A. Shapiro. "Asymptotic properties of statistical estimators in stochastic programming". In: *The Annals of Statistics* 17.2 (1989), pp. 841–858.

[88] J. Dupacová and R. Wets. "Asymptotic behavior of statistical estimators and of optimal solutions of stochastic optimization problems". In: *The annals of statistics* 16.4 (1988), pp. 1517–1549.

[89] L. Le Cam, L. M. LeCam, and G. L. Yang. *Asymptotics in statistics: some basic concepts*. Springer Science & Business Media, 2000.

[90] A. W. Van der Vaart. *Asymptotic statistics*. Vol. 3. Cambridge university press, 2000.

[91] J. C. Duchi and F. Ruan. "Asymptotic optimality in stochastic optimization". In: *The Annals of Statistics* 49.1 (2021), pp. 21–48.

[92] Y. Nesterov. "Primal-dual subgradient methods for convex problems". In: *Mathematical programming* 120.1 (2009), pp. 221–259.

[93] L. Xiao. "Dual averaging method for regularized stochastic learning and online optimization". In: *Advances in Neural Information Processing Systems* 22 (2009).

[94] L. Qi and J. Sun. "A nonsmooth version of Newton's method". In: *Mathematical programming* 58.1-3 (1993), pp. 353–367.

[95] R. Heckel and P. Hand. "Deep Decoder: Concise Image Representations from Untrained Non-convolutional Networks". In: *International Conference on Learning Representations*. 2018.

[96] P. Hand, O. Leong, and V. Voroninski. "Phase retrieval under a generative prior". In: *Advances in Neural Information Processing Systems* 31 (2018).

[97] P. Hand and V. Voroninski. "Global guarantees for enforcing deep generative priors by empirical risk". In: *Conference On Learning Theory*. PMLR. 2018, pp. 970–978.

[98] M. Asim, M. Daniels, O. Leong, A. Ahmed, and P. Hand. "Invertible generative models for inverse problems: mitigating representation error and dataset bias". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 399–409.

[99] E. J. Candes, X. Li, and M. Soltanolkotabi. "Phase retrieval via Wirtinger flow: Theory and algorithms". In: *IEEE Transactions on Information Theory* 61.4 (2015), pp. 1985–2007.

[100] J. Bolte, A. Daniilidis, and A. Lewis. "Tame functions are semismooth". In: *Mathematical Programming* 117.1-2 (2009), pp. 5–19.

[101] J. Keller, L. Dial, and contributors. *modded-nanogpt*. https://github.com/KellerJordan/modded-nanogpt. GitHub repository, accessed 31 Oct 2025. 2025.

[102] K. Jordan and collaborators. *Modded-NanoGPT (July 18 2025 snapshot)*. https://github.com/KellerJordan/modded-nanogpt/tree/0d20074260664590e2a1686b9371936944a3ddff. Commit 0d20074260664590e2a1686b9371936944a3ddff. 2025. (Visited on 07/18/2025).

[103] I. Shah, A. M. Polloreno, K. Stratos, P. Monk, A. Chaluvaraju, A. Hojel, A. Ma, A. Thomas, A. Tanwer, D. J. Shah, K. Nguyen, K. Smith, M. Callahan, M. Pust, M. Parmar, P. Rushton, P. Mazarakis, R. Kapila, S. Srivastava, S. Singla, T. Romanski, Y. Vanjani, and A. Vaswani. "Practical Efficiency of Muon for Pretraining". In: *arXiv preprint arXiv:2505.02222* (2025). DOI: 10.48550/arXiv.2505.02222.

[104] D. E. Carlson, E. Collins, C.-J. Hsieh, L. Carin, and V. Cevher. "Preconditioned Spectral Descent for Deep Learning". In: *Advances in Neural Information Processing Systems*. Vol. 28. 2015.

[105] D. Carlson, V. Cevher, and L. Carin. "Stochastic Spectral Descent for Restricted Boltzmann Machines". In: *International Conference on Artificial Intelligence and Statistics*. 2015, pp. 111–119.

[106] L. Xu, S. Xu, W. Xu, X. Xu, Y. Xu, Z. Xu, J. Yan, Y. Yan, X. Yang, Y. Yang, Z. Yang, Z. Yang, Z. Yang, H. Yao, X. Yao, W. Ye, Z. Ye, B. Yin, L. Yu, E. Yuan, H. Yuan, M. Yuan, H. Zhan, D. Zhang, H. Zhang, W. Zhang, X. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, Z. Zhang, H. Zhao, Y. Zhao, H. Zheng, S. Zheng, J. Zhou, X. Zhou, Z. Zhou, Z. Zhu, W. Zhuang, and X. Zu. "Kimi K2: Open Agentic Intelligence". In: *arXiv preprint arXiv:2507.20534* (2025).

[107] D. E. Carlson, E. Collins, Y.-P. Hsieh, L. Carin, and V. Cevher. "Preconditioned spectral descent for deep learning". In: *Advances in neural information processing systems* 28 (2015).